

Maria Teresa Figueiredo Gomes Ribeiro

Licenciatura em Matemática

**Utilização de Redes Neuronais num
Problema de Etiquetagem Gramatical**

Dissertação para obtenção do Grau de Mestre em
Matemática e Aplicações

Orientador: Rui Alberto Pimenta Rodrigues, Professor Auxiliar, Faculdade de Ciências e
Tecnologia, Universidade Nova de Lisboa

Júri:

Presidente: Prof. Doutor Jorge Orestes Cerdeira

Arguente : Prof. Doutor António José Rodrigues

Vogal: Prof. Doutor Rui Alberto Pimenta Rodrigues

Julho 2014

Utilização de Redes Neurais num Problema de Etiquetagem Gramatical

Copyright © Maria Teresa Figueiredo Gomes Ribeiro, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa. A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Resumo

As Redes Neurais têm aplicações numa grande diversidade de problemas, tendo a flexibilidade de poderem ser adaptadas a cada realidade em estudo. Nesta tese utilizam-se as redes neuronais como classificadores. As redes neuronais são, neste trabalho, aplicadas a um problema de linguagem natural. Construiu-se uma rede neuronal com duas camadas intermédias onde as funções de activação são a logística e a *softmax*. A função *softmax* foi utilizada para gerar os dados de saída (atribuição da probabilidade de uma palavra pertencer a cada classe).

O *software* usado para treinar a rede foi desenvolvido em Octave. Os dados processados basearam-se nos textos que constituem o Brown Corpus; estes textos foram decompostos em três grupos: treino, desenvolvimento e teste; por fim testou-se o desempenho da rede no conjunto de teste. Dos resultados obtidos podemos concluir que a rede aprende razoavelmente, registando-se uma percentagem de erro de 6.5% no conjunto de teste. No conjunto de treino obteve-se uma percentagem de erro inferior.

Palavras chave: Redes neuronais, linguagem natural, etiqueta gramatical, funções de activação

Abstract

Neural networks are applied in a great diversity of problems. Due to their flexibility they are adaptable to each reality in study. In this thesis the neural networks are used as classifiers. Neural networks are, in this work, applied to a problem of natural language. A neural network with two hidden layers was built where the activation functions used were the logistic function and the softmax function. The softmax function was used to generate the output (assignment of the probability of a word belonging to each class).

The software used to train the neural network was developed in Octave. The input was based in texts of the Brown Corpus. The texts were divided in three groups: training, development and test. Finally the performance of the network was tested in the test group. According to the results obtained we can conclude that the network learns reasonably well, the percentage of error was 6.5% in the test group. In the training group the percentage of error is inferior.

Keywords: Neural Networks, natural language, tag, activation functions

Agradecimentos

Agradeço, em primeiro lugar, ao meu orientador, Professor Doutor Rui Rodrigues por tudo o que me ensinou, pela sua total disponibilidade e infinita paciência.

Agradeço à São e à Dra Ana pela inexcedível ajuda, apoio e incentivo.

À Ana Isabel pela leitura atenta e incentivo.

Agradeço à minha família e aos meus amigos por todo o encorajamento.

Conteúdo

| | |
|---|-----------|
| Lista de figuras | ii |
| Lista de tabelas | v |
| 1 Introdução | 1 |
| 2 Processamento de Linguagem Natural e Etiquetagem gramatical | 3 |
| 2.1 Corpus Linguísticos | 3 |
| 2.2 Etiquetagem gramatical de palavras | 5 |
| 3 Tipos de Redes Neurais e de Funções de ativação | 9 |
| 3.1 Tipos de Redes Neurais | 9 |
| 3.2 Tipo de funções de ativação | 13 |
| 3.3 Perceptrons. A primeira geração de redes neurais | 14 |
| 3.4 Teorema da aproximação universal | 16 |
| 3.5 Atualização dos pesos | 18 |
| 4 Descrição do programa e Resultados obtidos | 21 |
| 4.1 Criação do vetor de entradas para a rede neuronal | 22 |
| 4.2 Inicialização dos pesos e <i>bias</i> de uma Rede Neuronal | 24 |
| 4.3 <i>Backprop</i> e função erro | 25 |
| 4.4 Resultados | 28 |
| 5 Conclusão | 31 |
| A Lista e significado das etiquetas do <i>Brown Corpus</i> | 35 |
| B Lista de etiquetas do <i>Brown Corpus</i> incluindo as etiquetas compostas | 51 |

| | | |
|----------|--|-----------|
| C | Ficheiro wlistnorarenonumbrown_train_dev_test.txt | 53 |
|----------|--|-----------|

Lista de Figuras

| | | |
|------|--|----|
| 2.1 | Categorias em vários Corpus Linguísticos incluindo na primeira coluna o <i>Brown Corpus</i> (ver [10]) | 4 |
| 3.1 | Unidades constituintes de uma rede neuronal (ver [6]) | 10 |
| 3.2 | Diagrama de uma rede neuronal com uma camada intermédia | 11 |
| 3.3 | Redes Neurais <i>Feed-forward</i> | 12 |
| 3.4 | <i>Recurrent Networks</i> | 12 |
| 3.5 | <i>Symmetrically Connected Networks</i> | 13 |
| 3.6 | Perceptron | 15 |
| 3.7 | Prova geométrica | 16 |
| 3.8 | Gráfico da função seno com erro 0.0081856. Gráfico pretendido a azul e gráfico dado pela rede a verde | 17 |
| 3.9 | Gráfico da função seno com erro 0.00041299. Gráfico pretendido a azul e gráfico dado pela rede a verde | 17 |
| 3.10 | Gráfico da função seno com erro 0.0028686. Gráfico pretendido a azul e gráfico dado pela rede a verde | 18 |
| 3.11 | Gráfico da função erro $E(w)$ | 19 |
| 4.1 | Rede neuronal com duas camadas intermédias usada neste trabalho. | 22 |
| 4.2 | Dados de entrada para a rede | 24 |
| 4.3 | % de palavras mal classificadas em função do número de épocas no conjunto de desenvolvimento | 29 |
| 4.4 | Evolução da função <i>cross-entropy</i> em função do número de épocas no conjunto de treino | 30 |
| 4.5 | % de palavras mal classificadas em função do número de épocas no conjunto de treino | 30 |
| A.1 | Etiquetas e seus significados | 36 |

| | | |
|------|--|----|
| A.2 | Etiquetas e seus significados. (Continuação) | 37 |
| A.3 | Etiquetas e seus significados. (Continuação) | 38 |
| A.4 | Etiquetas e seus significados. (Continuação) | 39 |
| A.5 | Etiquetas e seus significados. (Continuação) | 40 |
| A.6 | Etiquetas e seus significados. (Continuação) | 41 |
| A.7 | Etiquetas e seus significados. (Continuação) | 42 |
| A.8 | Etiquetas e seus significados. (Continuação) | 43 |
| A.9 | Etiquetas e seus significados. (Continuação) | 44 |
| A.10 | Etiquetas e seus significados. (Continuação) | 45 |
| A.11 | Etiquetas e seus significados. (Continuação) | 46 |
| A.12 | Etiquetas e seus significados. (Continuação) | 47 |
| A.13 | Etiquetas e seus significados. (Continuação) | 48 |
| A.14 | Etiquetas e seus significados. (Continuação) | 49 |
| A.15 | Etiquetas e seus significados. (Continuação) | 50 |

Lista de Tabelas

| | | |
|-----|--|---|
| 2.1 | O número de tipos de palavras no Brown Corpus(de acordo com DeRose, 1988). A palavra com 7 etiquetas é still | 6 |
|-----|--|---|

Capítulo 1

Introdução

O presente trabalho tem como objetivo a utilização das redes neuronais como classificadores. Neste trabalho as redes neuronais são aplicadas a um problema de linguagem natural, mais concretamente, são utilizadas para fazer a etiquetagem gramatical das palavras (*postagging*).

A etiquetagem gramatical das palavras consiste na atribuição de uma categoria gramatical a uma palavra, isto é, a cada palavra é atribuída, por exemplo, a categoria gramatical nome, verbo, adjetivo, advérbio, etc.

No presente trabalho utiliza-se uma rede neuronal *feed-forward* com duas camadas intermédias e usa-se como funções de ativação a função logística e a função *softmax*. O treino das redes neuronais é feito utilizando os textos do Corpus Linguístico *Brown Corpus*. Existem alguns trabalhos relacionados com a etiquetagem gramatical de palavras no Corpus Linguístico Português mas a existência de muitos trabalhos desenvolvidos nesta área utilizando o *Brown Corpus* e o facto de este ter sido o primeiro Corpus linguístico a ser utilizado por um computador foram as razões da sua escolha para este trabalho.

Desenvolve-se um procedimento para obter a etiquetagem gramatical das palavras que se inicia pela construção do vetor de entradas para a rede neuronal, ao que se segue o cálculo dos pesos iniciais e, por fim, um processo iterativo que termina quando se atingir um determinado critério de paragem. Neste caso o critério de paragem é o número de épocas igual a 100 (uma época corresponde a percorrer uma vez o conjunto de aprendizagem). Neste processo iterativo calcula-se em cada iteração a função erro e atualizam-se os pesos utilizando o algoritmo *backpropagation*. A escolha deste algoritmo deve-se ao facto de ser uma das técnicas mais eficientes para o cálculo da função gradiente. A função erro utilizada neste trabalho é a função *cross-entropy*.

Neste primeiro capítulo faz-se uma breve introdução ao trabalho desenvolvido o qual tem por obje-

tivo a utilização das redes neuronais como classificadores. No capítulo 2 apresentam-se sucintamente alguns Corpus linguísticos, apresenta-se um breve apontamento sobre a etiquetagem gramatical das palavras e referem-se alguns métodos usados na etiquetagem gramatical das palavras. No capítulo 3 faz-se uma descrição dos vários tipos de redes neuronais e dos tipos de funções de ativação. Descreve-se a primeira geração de redes neuronais, os perceptrons, e as suas limitações. Apresenta-se o teorema da aproximação universal e verifica-se que existe aproximação para a função contínua seno usando redes neuronais e uma função de ativação, a função logística. Apresenta-se ainda a atualização dos pesos considerando o vetor gradiente. No capítulo 4 faz-se uma descrição detalhada do programa desenvolvido para obter a etiquetagem gramatical das palavras utilizando as redes neuronais como classificadores e apresentam-se os resultados obtidos. Finalmente no capítulo 5 apresentam-se as conclusões.

Capítulo 2

Processamento de Linguagem Natural e Etiquetagem gramatical

O processamento da linguagem natural sofreu uma grande evolução, principalmente devido à cada vez maior necessidade de processamento de páginas *web* e à disponibilização no circuito comercial de técnicas de reconhecimento de fala.

É importante mostrar como os algoritmos e técnicas relacionados com a linguagem se podem aplicar a problemas do mundo real, tais como análise da ortografia, procura de palavras num documento, processamento de páginas *web*, *part-of-speech tagging* (etiquetagem gramatical de palavras) e tradução automática.

Neste trabalho utilizam-se as redes neuronais como classificadores. Pretende-se descobrir, dada uma sequência de palavras, a etiqueta gramatical (*tag*) da quarta palavra. Para tal criou-se um programa usando redes neuronais, as quais são treinadas com um conjunto de textos, o *Brown Corpus*.

Neste capítulo referem-se alguns Corpus Linguísticos que representam algumas linguagens nacionais entre os quais o *Brown Corpus*. Define-se etiquetagem gramatical e refere-se a dificuldade de resolver as ambiguidades na etiquetagem gramatical das palavras nas várias linguagens. Enumeram-se ainda alguns métodos usados na etiquetagem gramatical das palavras.

2.1 Corpus Linguísticos

Um Corpus linguístico pode ser definido como "um conjunto de textos ou partes de textos sobre os quais é feita uma análise linguística geral" (Meyer, 2002).

| Code | Text category | Brown | Frown | LOB | FLOB | Lancaster 1931 | Kolhapur | ACE | WWC | LCMC |
|------|---|-------|-------|-----|------|----------------|----------|-----|-----|------|
| A | Press reportage | 44 | 44 | 44 | 44 | 44 | 44 | 44 | 44 | 44 |
| B | Press editorials | 27 | 27 | 27 | 27 | 27 | 27 | 27 | 27 | 27 |
| C | Press reviews | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 17 |
| D | Religion | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 17 | 17 |
| E | Skills, trades and hobbies | 36 | 36 | 38 | 38 | 38 | 38 | 38 | 38 | 38 |
| F | Popular lore | 48 | 48 | 44 | 44 | 44 | 44 | 44 | 44 | 44 |
| G | Bio-graphies and essays | 75 | 75 | 77 | 77 | 77 | 70 | 77 | 77 | 77 |
| H | Miscellaneous (reports, official documents) | 30 | 30 | 30 | 30 | 30 | 37 | 30 | 30 | 30 |
| J | Science (academic prose) | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 | 80 |
| K | General fiction | 29 | 29 | 29 | 29 | 29 | 59 | 29 | 29 | 29 |
| L | Mystery and detective fiction | 24 | 24 | 24 | 24 | 24 | 24 | 15 | 24 | 24 |
| M | Science fiction | 6 | 6 | 6 | 6 | 6 | 2 | 7 | 6 | 6 |
| N | Western and adventure fiction | 29 | 29 | 29 | 29 | 29 | 15 | 8 | 29 | 29 |
| P | Romantic fiction | 29 | 29 | 29 | 29 | 29 | 18 | 15 | 29 | 29 |
| R | Humour | 9 | 9 | 9 | 9 | 9 | 9 | 15 | 9 | 9 |
| S | Historical fiction | — | — | — | — | — | — | 22 | — | — |
| W | Women's fiction | — | — | — | — | — | — | 15 | — | — |

Figura 2.1: Categorias em vários Corpus Linguísticos incluindo na primeira coluna o *Brown Corpus* (ver [10])

A maior parte dos países tem Corpus linguísticos que representam a linguagem nacional. Um Corpus linguístico ideal deveria cobrir a linguagem escrita e falada em proporções iguais. No entanto, a maioria dos Corpus linguísticos consistem apenas na informação relativa à linguagem escrita, uma vez que, a informação relativa à linguagem falada é mais difícil de obter.

O primeiro e mais conhecido Corpus linguístico que representa uma linguagem nacional é o British National Corpus (BNC). O BNC tenta ser o mais abrangente possível no que se refere ao Inglês Moderno. É constituído por 100 milhões de palavras, 90% correspondentes a textos escritos e 10% a linguagem falada.

O American National Corpus (ANC) iniciou-se em 1998 com o objetivo de construir um Corpus

linguístico nacional comparável ao BNC, mas para o inglês americano.

Existe também um Corpus linguístico nacional para o português. Este Corpus do Português contém mais de 45 milhões de palavras em mais de 50000 textos em português do século XIV ao século XX.

Mesmo antes da era do computador já existiam Corpus linguísticos constituídos por textos que eram analisados manualmente. Esta análise destinava-se a determinados projetos tais como, concordâncias bíblicas, elaboração de gramáticas e dicionários. Este trabalho era muito moroso e algo intediante. O primeiro Corpus linguístico utilizado por um computador foi o *Brown Corpus*.

O *Brown Corpus* foi desenvolvido na Brown University por Henry Kučera e Nelson Francis na década de 60 para o inglês americano. As amostras para este Corpus foram retiradas de textos escritos em inglês e publicados nos Estados Unidos da América a partir de uma lista de livros e periódicos existentes na biblioteca da Brown University e no Providence Athenaeum. Os textos foram agrupados em 15 categorias (ver figura 2.1), donde se extraíram 500 amostras, cada uma das quais com aproximadamente 2000 palavras (o número de palavras não é sempre 2000 uma vez que a amostra tem que ser constituída por frases completas) retiradas proporcionalmente de cada uma das categorias e perfazendo um total de cerca de 1,000,000 de palavras em inglês.

Os textos do *Brown Corpus* foram usados, neste trabalho, para treinar as redes neuronais. Existem alguns trabalhos relativos à etiquetagem gramatical de palavras no Corpus Português (ver [3]) mas o facto de existirem muitos trabalhos nesta área que utilizam o *Brown Corpus* fez-nos optar por este Corpus linguístico.

2.2 Etiquetagem gramatical de palavras

As palavras são tradicionalmente agrupadas em classes de equivalência designadas por *parts of speech* (classificações gramaticais), *pars orationis* (Latim), classes de palavras, classes morfológicas ou etiquetas lexicográficas. Nas gramáticas tradicionais em geral existem apenas algumas classificações gramaticais (substantivo, verbo, adjetivo, preposição, advérbio, conjunção, etc.). Nos modelos mais recentes existe um número muito maior de classes de palavras, por exemplo, 45 para o *Penn Treebank* (Marcus *et al.*, 1993), 87 para o *Brown Corpus* (Francis, 1979; Francis e Kučera, 1982) e 146 para o *C7* (Garside *et al.*, 1997).

Num Corpus linguístico, estudo da linguagem baseado em amostras (corpora) de textos do "mundo real", a etiquetagem gramatical das palavras (*part-of-speech tagging*, *POS tagging* ou *POST*) é o processo de atribuir a uma palavra num texto (corpus) uma categoria gramatical, como por exemplo, nome, verbo, adjetivo, advérbio, etc.

A etiquetagem gramatical utiliza-se num grande número de aplicações. É o primeiro passo numa análise sintática. É necessária quando se pretende fazer uma lematização correta das palavras. Utiliza-se ainda, por exemplo, para extrair informação de textos e na pesquisa lexicográfica.

Uma etiqueta gramatical (*tag*) é um caracter ou conjunto de caracteres que classifica gramaticalmente uma palavra.

Exemplo:

The/at Theatre-by-the-Sea/np ./, Matunuck/np ./, presents/vbz “/“ King/nn-tl Of/in-tl Hearts/nns-tl ”/” by/in Jean/np Kerr/np and/cc Eleanor/np Brooke/np ./.

at significa que a palavra é um artigo

np significa que a palavra é um nome próprio

vbz significa que a palavra é um verbo

As etiquetas também são atribuídas aos sinais de pontuação.

Os dados de entrada para um algoritmo de atribuição de etiquetas são uma sequência de palavras e um conjunto específico de etiquetas (por exemplo o conjunto de etiquetas associado ao *Brown Corpus*). A saída é a melhor etiqueta para a palavra.

Uma mesma palavra pode ter diferentes etiquetas.

| | |
|------------------------------|-------|
| Sem ambiguidade (1 etiqueta) | 36340 |
| Ambíguas (2-7 etiquetas) | 4100 |
| 2 etiquetas | 3760 |
| 3 etiquetas | 264 |
| 4 etiquetas | 61 |
| 5 etiquetas | 12 |
| 6 etiquetas | 2 |
| 7 etiquetas | 1 |

Tabela 2.1: O número de tipos de palavras no Brown Corpus(de acordo com DeRose, 1988). A palavra com 7 etiquetas é still

Apesar da maioria das palavras do *Brown Corpus* ter apenas uma etiqueta (só 11.5% das palavras têm mais do que uma etiqueta, ver tabela 2.1) a atribuição de uma etiqueta tem alguma ambiguidade.

Por exemplo, na frase ”O António casa com a Mariana no próximo Verão”a palavra *casa* é uma forma verbal, enquanto que em ”O António tem uma casa à beira-mar”, a palavra *casa* é um substantivo.

Mesmo numa frase simples como a anterior a atribuição de uma etiqueta tem alguma dificuldade. Tal também se verifica na língua inglesa. Por exemplo: em "In the still of the night", a palavra *still* é um substantivo, em "Be still" é uma forma verbal e em "He is still working" é uma preposição.

O objetivo da etiquetagem gramatical das palavras é solucionar estas ambiguidades, ou seja, ser capaz de atribuir a etiqueta gramatical certa de acordo com o contexto.

O trabalho de colocar as correspondentes etiquetas gramaticais no *Brown Corpus* foi inicialmente feito por Greene e Rubin. O programa obteve cerca de 70% de correspondências corretas. Estes resultados foram repetidamente revistos e corrigidos à mão até que, no final dos anos 70, o processo de etiquetagem gramatical estava quase perfeito. Estes textos foram usados como dados de aprendizagem da rede neuronal desenvolvida. Pretende-se que a rede seja capaz de atribuir a etiqueta gramatical certa consoante o contexto.

Muitos algoritmos para a etiquetagem gramatical das palavras têm vindo a ser desenvolvidos ao longo dos anos. Vários métodos são utilizados:

- Cadeias de Markov (Church, 1988; Cutting et al., 1992; Brants, 2000). Dentro deste grupo um dos algoritmos mais eficiente é o algoritmo Viterbi (Manning and Schütze, 1999).
- Modelos log-lineares.
- Redes Neurais.
- Árvores de decisão.
- Regras definidas manualmente para resolver a ambiguidade de certas palavras.

No trabalho apresentado nesta tese desenvolveu-se um programa usando redes neuronais treinadas com um conjunto de textos, o *Brown Corpus*.

Capítulo 3

Tipos de Redes Neurais e de Funções de ativação

O termo rede neuronal tem origem nas tentativas de encontrar uma representação matemática da informação processada por um sistema biológico (McCulloch and Pitts, 1943; Widrow and Hoff, 1960; Rosenblatt, 1962; Rumelhart et al., 1986).

Algumas das aplicações são, por exemplo:

- Aproximação de funções, análise de regressão, tal como previsão e modelação de séries temporais.
- Classificação, incluindo reconhecimento de padrões e de sequências.
- Processamento de dados, incluindo *filtering, clustering, blind signal separation and compression*.

Existem vários tipos de redes neurais. Neste capítulo faremos uma breve descrição de alguns tipos bem como de algumas das funções de ativação mais usadas.

3.1 Tipos de Redes Neurais

Nesta secção apresentam-se de modo sucinto alguns tipos de redes neurais. As redes neurais são constituídas por unidades de processamento de informação, neurónios, como representado na figura 3.1. As unidades que constituem uma rede neuronal têm vários dados de entrada (*inputs*) e vários dados de saída (*outputs*).

Os elementos básicos de um modelo de redes neurais são:

- Sinapses ou Ligações
Cada uma é caracterizada por um peso.
- *Adder*
Calcula a soma dos dados de entrada (*inputs*) ponderada pelos pesos
- Função de ativação
Estabelece intervalos para a amplitude do output de uma unidade
- *Bias*
Permite aumentar ou diminuir o input da função de ativação.

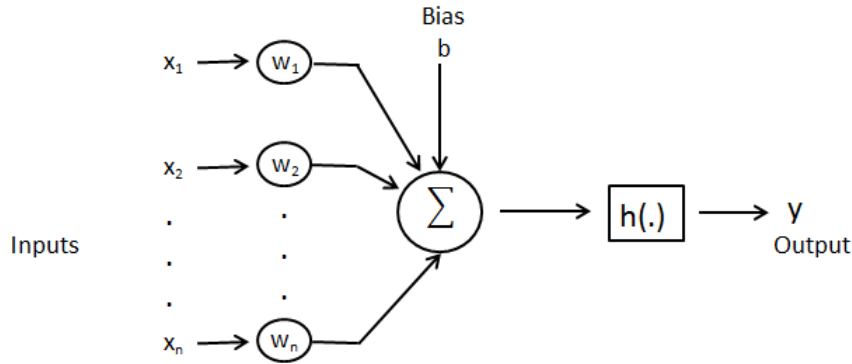


Figura 3.1: Unidades constituintes de uma rede neuronal (ver [6])

Para obter uma rede neuronal com uma camada intermédia (ver figura 3.2) constroem-se, em primeiro lugar, M combinações lineares das variáveis de entrada x_1, \dots, x_D , da forma

$$a_j = \sum_{i=1}^D w_{ji}^{(1)} x_i + w_{j0}^{(1)} \quad (3.1)$$

onde $j = 1, \dots, M$ e o índice (1) indica que os respectivos parâmetros se encontram na primeira camada da rede. Chamamos aos parâmetros $w_{ji}^{(1)}$ pesos e aos parâmetros $w_{j0}^{(1)}$ *bias*. A a_j chamamos ativação.

Cada um dos a_j é depois transformado usando uma função de ativação $h(\cdot)$ diferenciável e não linear dando origem a

$$z_j = h(a_j) \quad (3.2)$$

Os z_j designam-se por unidades ocultas ou intermédias. A função não linear $h(\cdot)$ é em geral uma função sigmóidal, tal como por exemplo, a função logística ou a função tangente hiperbólica.

Tal como em (3.1) estes valores dão origem a K unidades de output

$$a_k = \sum_{j=1}^M w_{kj}^{(2)} z_j + w_{k0}^{(2)} \quad (3.3)$$

onde $k = 1, \dots, K$ e K é o número total de outputs. Obtém-se assim a segunda camada da rede. Tal como anteriormente os $w_{k0}^{(2)}$ representam os *bias*.

Finalmente, a unidade de output é transformada, usando uma função de ativação apropriada, dando origem ao conjunto dos outputs da rede, y_k . A escolha da função de ativação é determinada pela natureza dos dados podendo ser, por exemplo, a identidade, a função logística ou a função *softmax*. A função geral de uma rede toma a forma

$$y_k(x, w) = h\left(\sum_{j=1}^M w_{kj}^{(2)} h\left(\sum_{i=1}^D w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right) \quad (3.4)$$

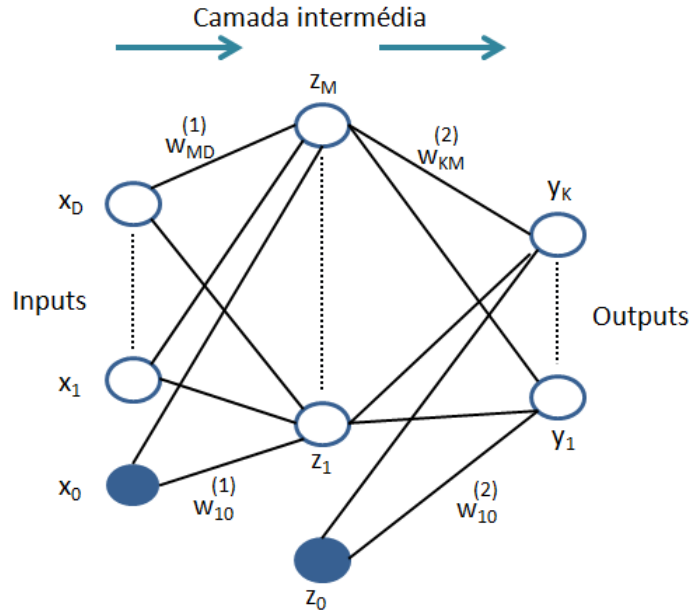


Figura 3.2: Diagrama de uma rede neuronal com uma camada intermédia

Apresentam-se de seguida alguns tipos de redes neuronais.

1. Redes Neurais *Feed-forward*

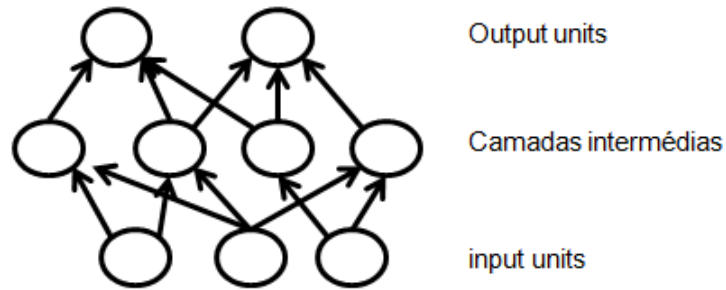


Figura 3.3: Redes Neurais *Feed-forward*

Estas redes neuronais são as mais comuns em aplicações práticas.

No caso de terem mais do que uma camada intermédia chamam-se *deep neural networks*.

2. *Recurrent Networks*

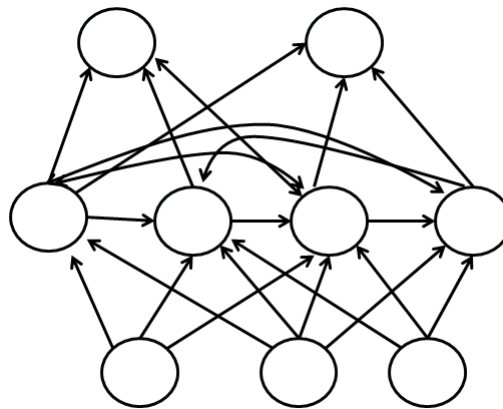
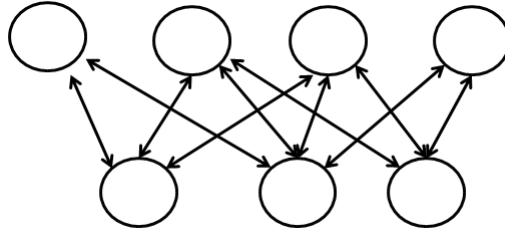


Figura 3.4: *Recurrent Networks*

Estas redes são muito potentes mas muito difíceis de treinar.

3. *Symetrically Connected Networks*Figura 3.5: *Symetrically Connected Networks*

Estas redes são semelhantes às *recurrent networks* mas têm ligações nos dois sentidos.

3.2 Tipo de funções de ativação

Uma função de ativação define o output de uma unidade. Existem várias funções de ativação. Apresentam-se seguidamente alguns dos tipos de função de ativação mais usados.

1. Funções Lineares (Unidades lineares)

As unidades em que se consideram funções de ativação lineares, $y = b + \sum_i x_i w_i$, são simples mas computacionalmente limitadas

2. Função *Binary Threshold* (Unidade *Binary Threshold*)

Existem duas maneiras equivalentes de escrever as equações para uma unidade *Binary Threshold*

$$z = \sum_i x_i w_i$$

$$y = \begin{cases} 1 & \text{se } z \geq \theta \\ 0 & \text{caso contrário} \end{cases}$$

ou

$$z = b + \sum_i x_i w_i$$

$$y = \begin{cases} 1 & \text{se } z \geq 0 \\ 0 & \text{caso contrário} \end{cases}$$

$$\theta = -b$$

3. Funções lineares *threshold* (Unidades lineares retificadas)

$$z = b + \sum_i x_i w_i$$

$$y = \begin{cases} z & \text{se } z \geq 0 \\ 0 & \text{caso contrário} \end{cases}$$

O output é uma função não linear do input total.

4. Funções sigmóidais

$$z = b + \sum_i x_i w_i$$

$$y = \frac{1}{1 + e^{-z}}$$

Geralmente o output, y , é dado pela função logística. Esta função é facilmente derivável o que torna a aprendizagem mais simples.

5. Funções estocásticas binárias

$$z = b + \sum_i x_i w_i$$

$$p(s = 1) = \frac{1}{1 + e^{-z}}$$

Usam as mesmas equações das unidades logísticas mas tratam o output como uma probabilidade.

6. Função *softmax*

$$y_k(x_i, w) = \frac{e^{\frac{a_k(x_i, w)}{\sum_j a_j(x_j, w)}}}{\sum_j e^{\frac{a_j(x_j, w)}{\sum_j a_j(x_j, w)}}}$$

em que $y_k \in [0, 1]$ e $\sum_k y_k = 1$

3.3 Perceptrons. A primeira geração de redes neurais

O perceptron (ver figura 3.6) é a rede neuronal mais simples (apenas uma camada).

Prova-se facilmente que um conjunto finito de amostras pode ser corretamente classificado utilizando um perceptron se e só se for linearmente separável, isto é, padrões com tipo diferente encontram-se nos lados opostos de um hiperplano. Por exemplo, as funções booleanas *and* e *or* podem ser executadas por um perceptron usando, por exemplo, a função de ativação *threshold* mas a função *xor* (ou exclusivo) não pode, tal como se ilustra seguidamente. As unidades dos perceptrons

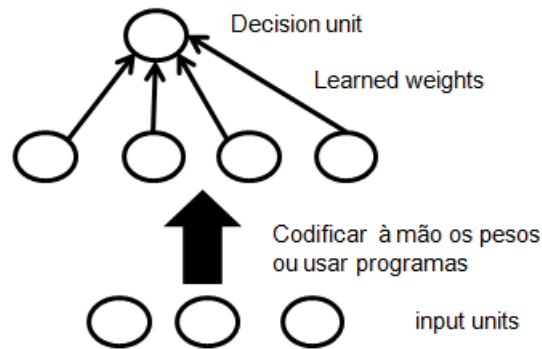


Figura 3.6: Perceptron

são neurónios *binary threshold*. Uma unidade *binary threshold* não consegue dizer se dois bits são o mesmo.

Temos então:

Casos Positivos (o mesmo byte) $(1, 1) \rightarrow 1; (0, 0) \rightarrow 1$

Casos Negativos (bits diferentes) $(1, 0) \rightarrow 0; (0, 1) \rightarrow 0$

Os quatro pares input-output dão origem a quatro desigualdades que são impossíveis de satisfazer.

Dos casos positivos obtemos

$$w_1 + w_2 \geq \theta \qquad 0 \geq \theta \qquad (3.5)$$

Dos casos negativos obtemos

$$w_1 < \theta \qquad w_2 < \theta \qquad (3.6)$$

Somando as inequações em (3.5) obtém-se $w_1 + w_2 \geq 2\theta$ e somando as inequações em (3.6) obtém-se $w_1 + w_2 < 2\theta$. Donde se conclui que as quatro inequações são impossíveis de satisfazer.

Este resultado pode também provar-se geometricamente

Considere um espaço no qual os eixos correspondem a componentes de um vetor de input.

- Cada vetor de input é um ponto neste espaço;
- Um vetor de pesos define um plano nesse espaço;
- O plano definido pelo vetor dos pesos é perpendicular ao vetor dos pesos e passa a uma distância da origem igual ao *threshold*

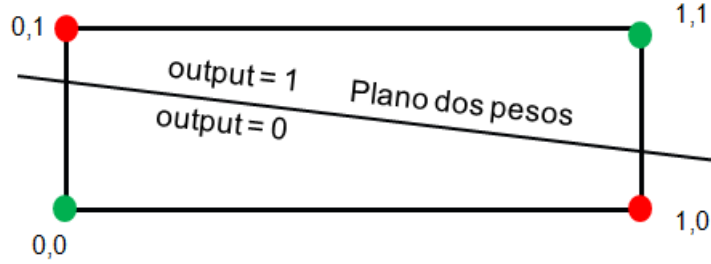


Figura 3.7: Prova geométrica

O caso positivo e o caso negativo não podem ser separados por um plano.

O processo de aprendizagem dos perceptrons não pode ser generalizado a uma rede com camadas intermédias. O processo de convergência dos perceptrons garante que cada vez que os pesos mudam estes se aproximam cada vez mais de um conjunto de pesos *generously feasible*. Esta garantia não se pode estender a redes mais complexas nas quais a média de duas boas soluções pode ser uma má solução.

3.4 Teorema da aproximação universal

O teorema da aproximação universal afirma que uma rede *feed-forward* com uma camada intermédia que contenha um número finito de unidades (neurónios) e com uma função de ativação limitada, monótona e contínua, φ , é um aproximador universal em $C(\mathbb{R}^m)$, isto é, nos compactos de \mathbb{R}^m .

Teorema 3.1. *Seja φ uma função de ativação limitada, monótona e contínua, $X \subseteq \mathbb{R}^m$ um conjunto compacto e $C(X)$ o espaço das funções contínuas em X .*

Então $\forall f \in C(X), \forall \varepsilon > 0, \exists n \in \mathbb{N}, a_{ij}, b_i, w_i \in \mathbb{R}, i \in \{1, \dots, n\}, j \in \{1, \dots, m\}$

tem-se $(A_n f)(x_1, \dots, x_n) = \sum_{i=1}^n w_i \varphi(\sum_{j=1}^m a_{ij} x_j + b_i)$ como uma aproximação da função $f(\cdot)$, isto é, $\|f - A_n f\| < \varepsilon$

Em $A_n f$, n representa o número de neurónios na camada intermédia.

Uma das primeiras versões do teorema da aproximação universal foi provada em 1989 por George Cybenko [8]. Kurt Honik (1991) demonstrou que o facto de redes deste tipo serem aproximadores universais deve-se principalmente à estrutura da rede e não tanto à função de ativação escolhida.

Para funções como o seno, uma função contínua, foi desenvolvido um programa considerando redes neurais e a função de ativação logística. Apresentam-se a título ilustrativo três gráficos obtidos para a função seno.

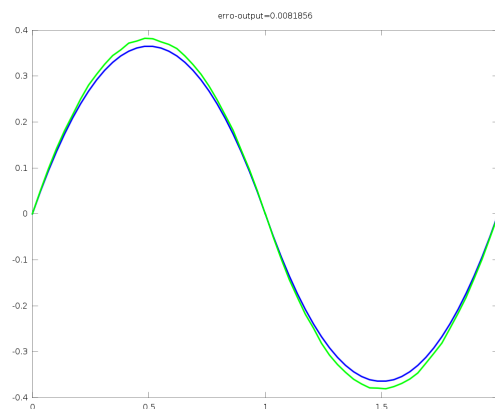


Figura 3.8: Gráfico da função seno com erro 0.0081856. Gráfico pretendido a azul e gráfico dado pela rede a verde

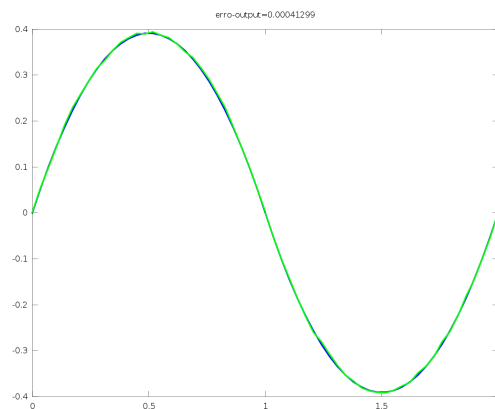


Figura 3.9: Gráfico da função seno com erro 0.00041299. Gráfico pretendido a azul e gráfico dado pela rede a verde

Para funções que não obedecem às hipóteses do teorema, como é o caso do problema em estudo, não se pode garantir a aproximação.

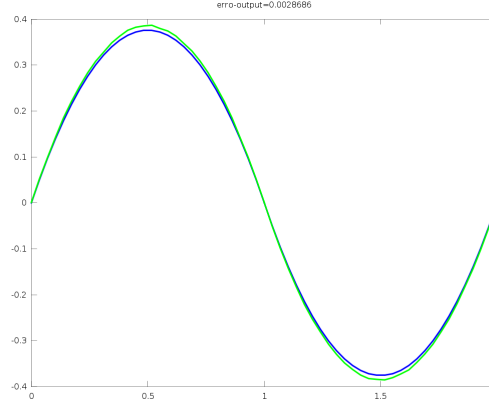


Figura 3.10: Gráfico da função seno com erro 0.0028686. Gráfico pretendido a azul e gráfico dado pela rede a verde

3.5 Atualização dos pesos

Uma rede neuronal pode ser considerada como uma classe genérica de funções não lineares de um vetor x , de variáveis de input num vetor y , de variáveis de output. Estas funções dependem de parâmetros desconhecidos tal como os pesos. Uma maneira de determinar os parâmetros da rede é minimizar a função erro dada pelo quadrado dos desvios. Dado um conjunto de treino composto por um conjunto de vetores de input x_n , com $n = 1, \dots, N$, e o correspondente conjunto de vetores que se pretende obter (vetores alvo), t_n , minimizamos a função erro

$$E(w) = \frac{1}{2} \sum_{n=1}^N \|y(x_n, w) - t_n\|^2 \quad (3.7)$$

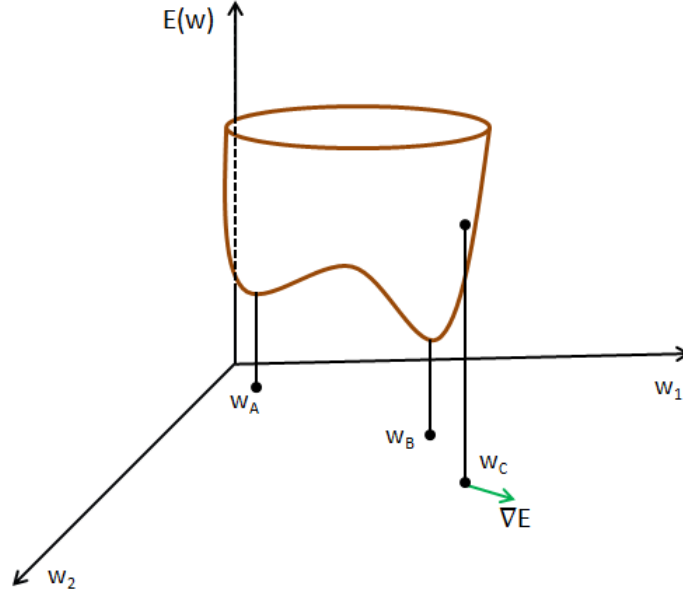
Pretendemos, deste modo, determinar o vetor dos pesos, w , que minimiza a função $E(w)$.

Note-se que se uma pequena deslocação de w para $w + \delta w$ a variação no erro será de

$$\delta E \simeq \delta w^T \nabla E(w)$$

onde o vetor $\nabla E(w)$ aponta na direção da maior crescimento da função erro.

Uma vez que, $E(w)$ é uma função de w contínua e diferenciável, o seu menor valor ocorre num ponto do espaço dos pesos onde o gradiente de $E(w)$ se anula. Dado que não é possível determinar uma solução analítica da equação $E(w) = 0$, teremos que utilizar processos numéricos. Na maioria

Figura 3.11: Gráfico da função erro $E(w)$

destas técnicas escolhe-se um valor inicial w^0 para o vetor dos pesos e em seguida atualizam-se os pesos numa sucessão de passos da forma

$$w^{\gamma+1} = w^{\gamma} + \Delta w^{\gamma} \quad (3.8)$$

onde γ representa o passo iterativo.

Existem diferentes maneiras de atualizar o vetor Δw^{γ} . Muitos algoritmos usam informação dada pelo vetor gradiente pelo que necessitam que após cada atualização o valor $\nabla E(w)$ seja calculado no novo vetor $w^{\gamma+1}$.

A maneira mais simples de utilizar a informação dada pelo vetor gradiente é escolher a atualização dos pesos em (3.8) que contenha uma pequena deslocação (passo) na direção do vetor gradiente e sentido contrário, isto é, na direção do gradiente negativo. Assim,

$$w^{\gamma+1} = w^{\gamma} - \eta \nabla E(w^{\gamma}) \quad (3.9)$$

onde o parâmetro positivo η se designa de taxa de aprendizagem. Após cada atualização, o vetor gradiente é calculado no novo vetor de pesos e assim sucessivamente.

Assim, é importante encontrar uma técnica eficiente para calcular o gradiente de uma função erro $E(w)$ para uma rede neuronal *feed-forward*. Uma das técnicas mais eficiente é o algoritmo

backpropagation. É este o algoritmo usado neste trabalho e que será descrito em detalhe no capítulo seguinte.

Capítulo 4

Descrição do programa e Resultados obtidos

Neste capítulo faz-se uma descrição detalhada do programa desenvolvido, usando redes neuronais, para dada uma sequência de palavras descobrir a etiqueta da quarta palavra.

A rede neuronal utilizada neste trabalho é uma rede *feed-forward* com duas camadas intermédias (ver figura 4.1). O número de inputs foi 3016. O número de elementos da primeira camada intermédia foi 4096 e o número de elementos da segunda camada intermédia foi também 4096. Utilizou-se a potência de 2 mais próxima do número de etiquetas, (472), isto é, o valor 512. O número de caracteres diferentes usados foi 90 e o número de palavras distintas 17575.

As funções de ativação utilizadas foram as seguintes:

- função logística: da camada de input para a primeira camada intermédia e da primeira camada intermédia para a segunda
- função *softmax*: da segunda camada intermédia para o output.

Utiliza-se a função *softmax* para atribuir a probabilidade de uma palavra pertencer a cada classe (o número de classes é igual ao número de etiquetas).

Os principais procedimentos do programa desenvolvido encontram-se no algoritmo 1.

No programa desenvolvido neste trabalho usa-se como critério de paragem o número máximo de épocas igual a 10 e continua-se o treino se não se verificar *overfitting*. Uma época corresponde a percorrer o conjunto de aprendizagem uma vez. Poder-se-ia ter considerado outros critérios de paragem, como por exemplo, uma majoração para o erro.

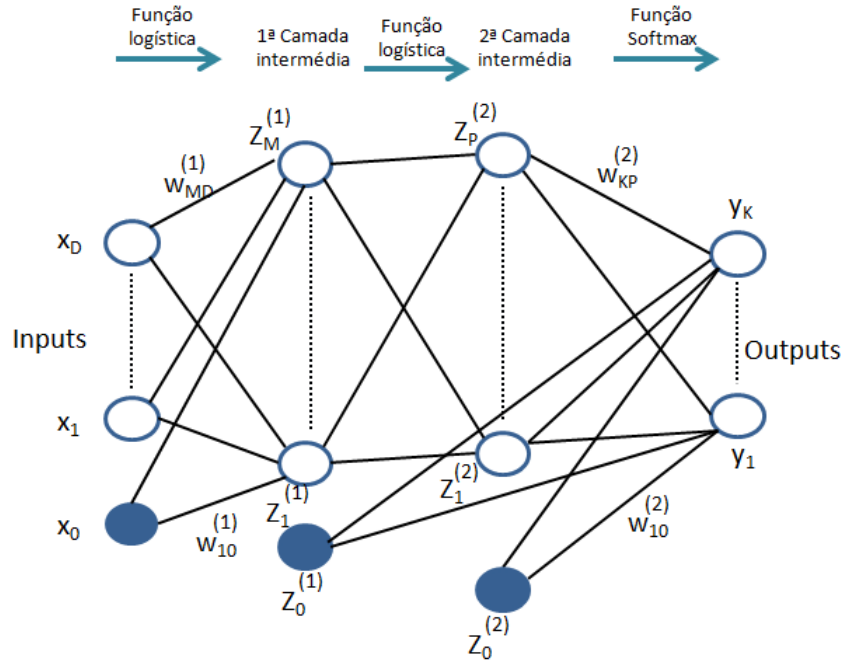


Figura 4.1: Rede neuronal com duas camadas intermédias usada neste trabalho.

4.1 Criação do vetor de entradas para a rede neuronal

Para criar o vetor de entradas para a rede neuronal começa-se por pegar nos textos que constituem o *Brown Corpus* e em cada texto separa-se a palavra da respetiva etiqueta. Cada texto origina dois ficheiros, um com as etiquetas e outro com as palavras. Cada ficheiro com as etiquetas é posteriormente convertido num outro ficheiro em que a cada etiqueta é atribuído um número de acordo com a correspondência indicada no ficheiro *browntagset.txt*. O conjunto de etiquetas neste ficheiro encontra-se incompleto, uma vez que não contém as etiquetas compostas, pelo que teve que

Algorithm 1 Método usado

Criação do vetor de entradas para a rede neuronal (ver figura 4.2)

Inicialização dos pesos e *bias* para iniciar o *backprop*

REPETIR

 Cálculo da função erro

Backprop (Atualização dos pesos)

ENQUANTO não se atingir o critério de paragem.

ser atualizado construindo um novo conjunto de etiquetas.

Do mesmo modo cada ficheiro com as palavras é convertido em dois ficheiros: um em que a cada palavra é atribuído um número de acordo com a correspondência indicada no ficheiro `wlistnorarenonumbrown_train_dev_test.txt` e outro onde consta o número da palavra na lista de todas as palavras do *Brown Corpus*.

Consideram-se duas listas associadas ao *Brown Corpus*, uma lista que se designa por LISTA 1 que contém todas as palavras do *Brown Corpus* e uma LISTA 2 que contém as palavras que aparecem mais do que três vezes e ainda as duas palavras NUMBER e UNKNOWN. As palavras que aparecem menos do que três vezes são convertidas em UNKNOWN e os números são convertidos em NUMBER. Convertem-se as palavras em vetores de zeros e um um. Como o número de palavras distintas é 17525 o vetor correspondente a cada uma das palavras tem dimensão 17525. Para uma determinada palavra todas as entradas do vetor são zero excepto a entrada relativa ao número atribuído à palavra de acordo com a correspondência indicada no ficheiro `wlistnorarenonumbrown_train_dev_test.txt` que tem valor um.

Por exemplo, a palavra *Academy*, a que corresponde o número 54 (ver anexo B onde se apresenta um extrato do ficheiro `wlistnorarenonumbrown_train_dev_test.txt`) é transformada num vetor com dimensão 17525 em que a entrada 54 é um e as restantes são zero.

As palavras que ocorrem mais do que três vezes, isto é, as palavras da LISTA 2, são convertidas em vetores de 100 números reais (ver [16]). A conversão das palavras em números reais possibilita que palavras semelhantes tenham uma representação semelhante em termos de vetores de reais (ver[9]). Assim, quando se faz o ajustamento de parâmetros num modelo, relativamente a uma palavra ou sequência de palavras este repercute-se nas palavras ou sequências de palavras semelhantes. São estes os vetores de entrada na rede correspondentes às palavras.

No seguimento da construção dos dados de entrada para a rede constrói-se um vetor em que as três primeiras entradas correspondem às três primeiras palavras, as três entradas seguintes às etiquetas das três primeiras palavras, a entrada seguinte corresponde à palavra que pretendemos aprender a classificar e as três seguintes entradas correspondem a três palavras. Existem ainda mais dez entradas que correspondem aos dez últimos caracteres (se existirem) da palavra que pretendemos classificar (ver figura 4.2).

A rede usada neste trabalho aprende que, por exemplo, a seguir a um nome o mais provável é aparecer uma forma verbal.

Na matriz dos dados de entrada cada vetor corresponde a uma linha e cada coluna corresponde a uma variável. A matriz dos dados de entrada deve ser normalizada para acelerar o processo de aprendizagem da rede. A normalização é feita transformando cada coluna numa variável com média

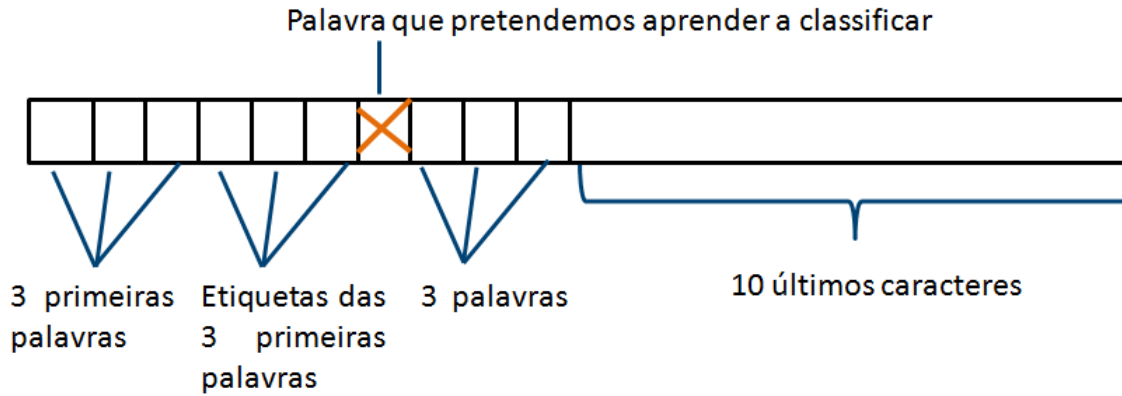


Figura 4.2: Dados de entrada para a rede

zero e desvio padrão 1.

4.2 Inicialização dos pesos e *bias* de uma Rede Neuronal

Numa rede neuronal com mais do que uma camada oculta, o algoritmo *backprop*, na sua versão tradicional, não permite uma correção adequada dos pesos entre as camadas mais afastadas da camada de saída, quando a inicialização destes é feita aleatoriamente. Impõe-se por isso uma inicialização mais cuidadosa destes pesos e *bias*.

Uma forma de obter uma inicialização que pode posteriormente ser afinada pelo *backprop* é usar uma sobreposição de *Restricted Boltzmann Machines* (RBMs).

Uma *Restricted Boltzmann Machines* é uma rede neuronal constituída por dois conjuntos de unidades, visíveis e ocultas. As ligações entre as duas camadas são bidireccionais e apenas existem entre unidades de camadas diferentes. As funções de activação usadas na camada oculta são binárias probabilísticas onde a probabilidade de o resultado ser um é dado pela função logística. No caso das unidades visíveis consideraremos funções de activação probabilísticas binárias como nas unidades ocultas ou a identidade afetada de ruído gaussiano.

Para utilizar uma RBM para inicializar os pesos e *bias* entre a camada de entrada de uma rede neuronal e a primeira camada oculta consideraremos o seguinte procedimento:

- Inicializamos os valores das unidades visíveis da RBM com um vetor de entrada na rede neuronal.
- Atualizamos os valores das unidades ocultas das RBM fixados os valores das unidades visíveis.

- Fixados os valores das unidades ocultas atualizamos os valores das unidades visíveis e apelidamos estes valores de reconstrução do vetor de entradas da rede neuronal.

Segundo um processo, apelidado de *Contrastive Divergence*, os pesos e *bias* podem ir sendo corrigidos de modo a reduzir a discrepância entre a distribuição dos vetores de entrada e as suas reconstruções. Os valores dos pesos desta RBM são usados para os pesos entre a camada de entrada da rede neuronal e a primeira camada oculta. Os *bias* da camada oculta da RBM são usados para *bias* das unidades da primeira camada oculta da rede neuronal.

Depois de inicializados os pesos e *bias* entre a camada de entrada e a primeira camada oculta da rede neuronal, estes são usados para determinar os valores da primeira camada oculta correspondentes a cada vetor de entrada na rede neuronal. O conjunto dos vetores correspondentes aos valores das unidades da primeira camada oculta podem ser considerados os vetores de entrada numa rede neuronal com menos uma camada oculta onde a camada de entrada corresponde à primeira camada oculta da rede inicial. Repetiremos o procedimento anterior para esta nova rede neuronal e desta forma são inicializados os pesos e *bias* da rede neuronal inicial, excepto aqueles entre a última camada oculta e a camada de saída. Estes últimos pesos podem ser inicializados aleatoriamente pois o *backprop* facilmente efectua as correções necessárias.

4.3 *Backprop* e função erro

O objetivo deste procedimento é encontrar um conjunto de pesos que garanta que, para cada vetor do input, o vetor output determinado pela rede seja o mais próximo possível do vetor output desejado. Se houver um número fixo e finito de casos input-output, o erro total da performance da rede com um determinado conjunto de pesos pode ser calculado comparando, em cada caso, o output dado pela rede e o output desejado.

No cálculo da função erro utiliza-se a função *cross-entropy*. No caso de haver apenas duas classes ($L=2$) a fórmula do erro $E(w)$ obtém-se do seguinte modo:

Para um conjunto de dados Φ_n, t_n onde $\Phi_n = \Phi(x_n)$ com, $n = 1, \dots, N$, a função de verossimilhança pode ser escrita como

$$p(t|w) = \prod_{i=1}^N y_n^{t_n} \{1 - y_n\}^{1-t_n}$$

onde $t = (t_1, \dots, t_N)^T$ e $y_n = p(C_1|\Phi_n)$.

Pode definir-se uma função erro considerando o logaritmo negativo da função de verossimilhança,

o que origina a função *cross-entropy* na forma

$$E(w) = -\ln p(t|w) = -\sum_{n=1}^N \{t_n \ln y_n + (1 - t_n) \ln(1 - y_n)\}$$

onde $y_n = \sigma(a_n)$ e $a_n = w^T \phi_n$, sendo

t_n : verdadeira classificação da palavra (1 se a palavra pertence à classe; 0 caso contrário)

Φ_n : classificação dada pela rede (probabilidade, atribuída pela rede, da amostra pertencer a uma das classes)

Para L classes o erro é dado por

$$E(w) = -\sum_{n=1}^N \left(\sum_{i=1}^L t_{n,i} \ln y_{n,i} \right) \quad (4.1)$$

Apresenta-se seguidamente uma dedução considerando L=3, isto é, três classes. A função *softmax* é neste caso dada por

$$\Phi(x) = \frac{e^x}{e^x + e^{x_2} + e^{x_3}}$$

Temos dois casos:

Caso 1 : $t_1 = 1$ e $t_2 = t_3 = 0$

$$F(x) = \frac{e^x}{e^x + e^{x_2} + e^{x_3}} = \frac{e^x}{e^x + a} = \frac{1}{1 + ae^{-x}}$$

$$1 - F(x) = \frac{ae^{-x}}{1 + ae^{-x}}$$

$$F'(x) = \frac{ae^{-x}}{(1 + ae^{-x})^2} = F(x)(1 - F(x))$$

$$E(x) = 1 \cdot \ln \frac{e^x}{e^x + e^{x_2} + e^{x_3}} = 1 \cdot \ln \Phi(x)$$

$$E'(x) = \frac{\Phi'(x)}{\Phi(x)} = \frac{\Phi(x)(1 - \Phi(x))}{\Phi(x)} = 1 - \Phi(x)$$

Caso 2 : $t_2 = 1$ e $t_1 = t_3 = 0$

$$G(x) = \frac{e^{x_2}}{e^x + e^{x_2} + e^{x_3}} = \frac{1}{e^{x-x_2} + b} = \frac{1}{b + ae^x}$$

onde $b = 1 + e^{x_3-x_2}$ e $a = e^{-x_2}$

$$1 - bG(x) = \frac{ae^x}{b + ae^x}$$

$$G'(x) = -\frac{ae^x}{(b + ae^x)^2} = -G(x)(1 - bG(x))$$

$$\begin{aligned}
E(x) &= 1. \ln \frac{e^{x_2}}{e^x + e^{x_2} + e^{x_3}} = 1. \ln G(x) \\
E'(x) &= \frac{G'(x)}{G(x)} = -(1 - bG(x)) = -\frac{ae^x}{b + ae^x} = \\
&= -\frac{e^{-x_2}e^x}{e^{-x_2}e^x + 1 + e^{-x_2}e^{x_3}} = -\frac{e^x}{e^x + e^{x_2} + e^{x_3}}
\end{aligned}$$

Para minimizar E , através do gradiente descendente, é necessário calcular a derivada de E em ordem a cada peso ou seja, a soma das derivadas parciais para cada caso input-output. O passo *backward* que propaga as derivadas começando na última camada até à primeira camada começa por calcular $\frac{\partial E}{\partial \Phi}$ para cada unidade de output. Derivando a equação (4.1) para um caso particular obtém-se

$$\frac{\partial E}{\partial \Phi_j} = \Phi_j - t_j \quad (4.2)$$

Aplicando a regra da cadeia podemos calcular $\frac{\partial E}{\partial x_j}$

$$\frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial \Phi_j} \frac{d\Phi}{dx_j} \quad (4.3)$$

Como $\Phi_j = \frac{1}{1+e^{-x_j}}$, derivando esta equação para obter o valor de $\frac{d\Phi}{dx_j}$ e substituindo em (4.3) obtém-se

$$\frac{\partial E}{\partial x_j} = \frac{\partial E}{\partial \Phi_j} \Phi_j (1 - \Phi_j) \quad (4.4)$$

Pela equação (4.4) ficamos a saber como é que uma variação no input total x afeta o erro. Como $x_j = \sum \Phi_i w_{ij}$ o input total é uma função linear de Φ e dos pesos. Assim, torna-se fácil determinar como o erro será afetado ao modificarmos Φ e w . Para um peso w_{ji} temos:

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial x_j} \frac{\partial x_j}{\partial w_{ji}} = \frac{\partial E}{\partial x_j} \Phi_j \quad (4.5)$$

e para o output da i -ésima unidade a contribuição para $\frac{\partial E}{\partial \Phi_j}$ é

$$\frac{\partial E}{\partial \Phi_j} = \frac{\partial E}{\partial x_j} \frac{\partial x_j}{\partial \Phi_j} = \frac{\partial E}{\partial x_j} w_{ij} \quad (4.6)$$

Assim, tendo em conta todas as conexões a partir da unidade i tem-se

$$\frac{\partial E}{\partial \Phi_i} = \sum_j \frac{\partial E}{\partial x_j} w_{ji} \quad (4.7)$$

Podemos deste modo calcular $\frac{\partial E}{\partial \Phi}$ para cada unidade na penúltima camada conhecendo $\frac{\partial E}{\partial \Phi}$ para todas as unidades na última camada. Este procedimento pode repetir-se para as camadas anteriores.

Atualizamos os pesos fazendo

$$\Delta w = -\epsilon \frac{\partial E}{\partial w} \quad (4.8)$$

onde ϵ é a taxa de aprendizagem (*learning rate*). Esta atualização pode ser feita após cada caso ou após um grupo (*batch*) de casos .

4.4 Resultados

Os resultados computacionais usando o programa descrito foram obtidos considerando os textos do *Brown Corpus*. Os textos deste Corpus linguístico foram decompostos em três grupos: treino, desenvolvimento e teste. Durante o treino foram usados os conjuntos de treino e desenvolvimento. No fim testa-se o desempenho de rede no conjunto de teste.

Em primeiro lugar correu-se o programa de treino da rede. Utilizaram-se 100 épocas, o que demorou cerca de 26 horas. Correu-se o programa na placa gráfica (200 processadores) usando a linguagem Cuda. Em Octave demoraria 2 semanas pelo que se converteu para Cuda. Ainda se correram em seguida mais 5 épocas mas os resultados obtidos pioraram devido a *overfitting*. O conjunto de ficheiros que constitui o grupo de desenvolvimento (ficheiros 'numpalca01.txt', 'numpalcc03.txt', 'numpalce05.txt', 'numpalcg07.txt', 'numpalci09.txt' e 'numpalck01.txt') foi usado apenas para controlar o *overfitting* durante o treino. O teste do desempenho da rede no conjunto de desenvolvimento foi feito no fim de cada época. Compararam-se os resultados obtidos e obteve-se deste modo informação sobre o momento de paragem do algoritmo *backprop*. Quando os resultados obtidos no conjunto de desenvolvimento começam a ser piores que os obtidos no conjunto de treino significa que há *overfitting* e deve parar-se o processo.

Na figura 4.3 apresenta-se a percentagem de palavras mal classificadas em função do número de épocas no conjunto de desenvolvimento.

Nas figuras 4.4 e 4.5 apresentam-se respetivamente, a evolução da função *cross-entropy* e da percentagem de palavras mal classificadas em função do número de épocas no conjunto de treino.

Testa-se finalmente o desempenho da rede no conjunto de teste. Utilizam-se os ficheiros 'numpalcb02.txt', 'numpalcd04.txt', 'numpalcf06.txt', 'numpalch08.txt', 'numpalcj10.txt' e 'numpalcl02.txt'. Todos os outros textos pertencem ao conjunto de treino da rede.

Os resultados obtidos para o conjunto de teste foram 759 erros em 11613 palavras, isto é, uma percentagem de erro de 6.5%.

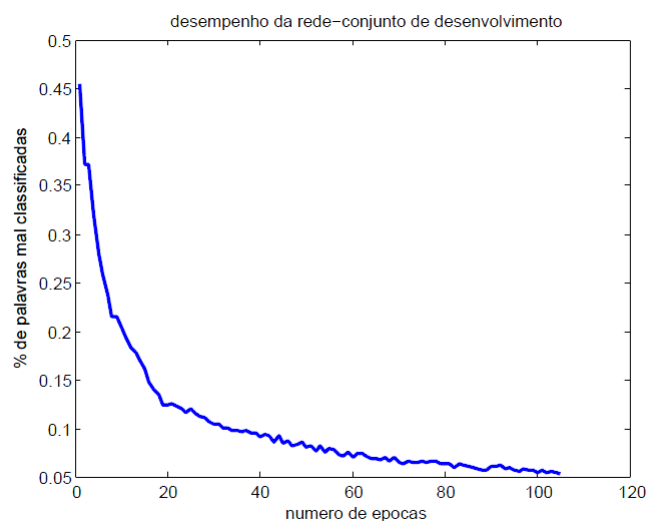


Figura 4.3: % de palavras mal classificadas em função do número de épocas no conjunto de desenvolvimento

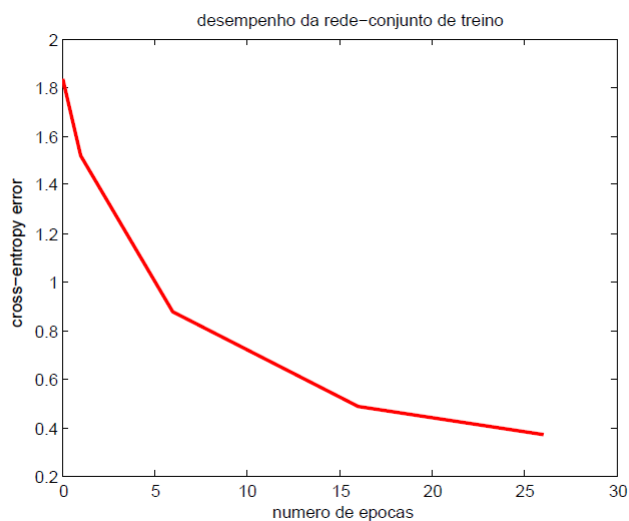


Figura 4.4: Evolução da função *cross-entropy* em função do número de épocas no conjunto de treino

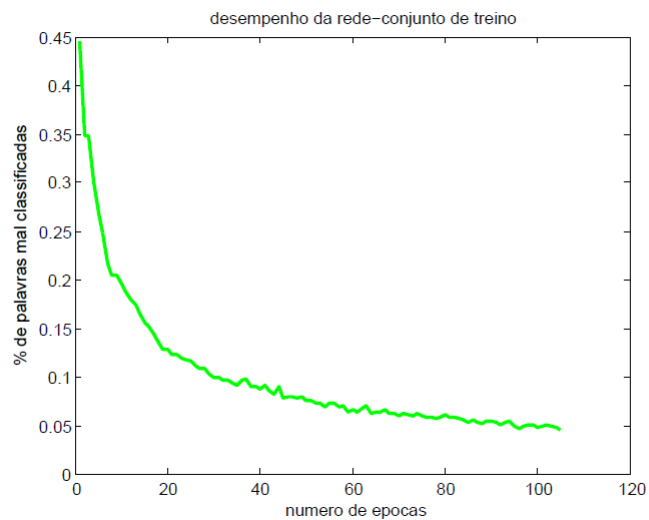


Figura 4.5: % de palavras mal classificadas em função do número de épocas no conjunto de treino

Capítulo 5

Conclusão

Neste trabalho utilizaram-se as redes neuronais como classificadores. As redes neuronais foram aplicadas a um problema de linguagem natural, mais concretamente, o problema de etiquetagem gramatical das palavras (*postagging*).

Utilizou-se uma rede neuronal *feed-forward* com duas camadas intermédias e consideraram-se as funções de ativação logística e *softmax*. O treino das redes neuronais foi feito recorrendo aos textos do Corpus Linguístico *Brown Corpus*. Apesar de haver alguns trabalhos desenvolvidos nesta área no Corpus Linguístico Português optou-se pelo *Brown Corpus*.

Para fazer a etiquetagem gramatical das palavras desenvolveu-se um procedimento em que a partir de um vetor de entradas inicial e de uns pesos iniciais se vão atualizando os pesos utilizando o algoritmo *backpropagation* de modo a que o erro obtido seja cada vez menor. A escolha do algoritmo *backpropagation* deve-se ao facto de ser uma das técnicas mais eficientes para o cálculo da função gradiente. A função erro utilizada neste trabalho foi a função *cross-entropy*.

A rede conseguiu não só memorizar as etiquetas gramaticais das palavras com uma só etiqueta mas também atribuir a etiqueta certa às palavras ambíguas num número razoável de casos.

Bibliografia

- [1] Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C. (2003), A Neural Probabilistic Language Model, *Journal of Machine Learning Research*, 3, 1137-1155.
- [2] Bishop, C. (2006), Pattern Recognition and Machine Learning, *Springer*.
- [3] Branco, A., Silva, J. (2004), Evaluating Solutions for the Rapid Development of State-of-the-Art POS Taggers for Portuguese. *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC2004), Paris, ELRA*, 507-510.
- [4] Brants, T. (2000), TnT - a statistical part-of-speech tagger, *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000, Seattle, WA*.
- [5] Church, K. W. (1988), A stochastic parts program and noun phrase parser for unrestricted text, *Proceedings of the Second Conference on Applied Natural Language Processing*, 136-143.
- [6] Csáji, B. C. (2001), Approximations with Artificial Neural Networks, Msc thesis.
- [7] Cutting, D., Kupiec, J., Pedersen, J., Sibun, P. (1992), A practical part-of-speech tagger, *Proceedings of the Third Conference on Applied Natural Language Processing, Trento, Italy*, 133-140.
- [8] Cybenko, G. (1989), Approximations by superpositions of sigmoid functions, *Mathematics of Control, Signals and Systems*, 2(4), 303-314.
- [9] Erk, K. (2009), Representing words as regions in vector space, *Proceedings of the 13th Conference on Computational Natural Language Learning*, 57-65.
- [10] *Handbooks of Linguistics and Communication Science* edited by Herbert Ernst Wilgard.
- [11] Hinton, G. (2012), Neural Networks for Machine Learning, *Courses*.
- [12] Hornik, K. (1991), Approximation capabilities for Multilayer Feedforward Networks, *Neural Networks*, 4(2), 251-257.

- [13] Ilya, S., Matens, J., Hinton, G. (2011), Generating Text with Recurrent Neural Networks, *Proceedings of the 28th International Conference on Machine Learning*.
 - [14] Manning, C. D., Schütze, H. (1999), Foundations of statistical natural language processing, *The MIT Press, Cambridge, Ma*.
 - [15] Meyer, C. F. (2002), English Corpus Linguistics: an Introduction, *Cambridge University Press*.
 - [16] Mnih, A., Hinton, G.E. (2008), A Scalable Hierarchical Distributed Language Model, *Proceedings of the Advances in Neural Information Processing Systems 21 Conference*, 1081-1088.
-

Apêndice A

Lista e significado das etiquetas do *Brown Corpus*

Neste anexo apresentam-se as etiquetas do *Brown Corpus* e respetivos significados tal como aparecem em <https://www.comp.leeds.ac.uk/ccalas/tagsets/brown.html>

| Tag | Description | Examples |
|--------------|--|---|
| (| opening parenthesis | (|
|) | closing parenthesis |) |
| * | negator | not n't |
| , | comma | , |
| -- | dash | -- |
| . | sentence terminator | . ? ; ! : |
| : | colon | : |
| ABL | determiner/pronoun, pre-qualifier | quite such rather |
| ABN | determiner/pronoun, pre-quantifier | all half many nary |
| ABX | determiner/pronoun, double conjunction or pre-quantifier | both |
| AP | determiner/pronoun, post-determiner | many other next more last former little several enough most least only very few fewer past same Last latter less single plenty 'nough lesser certain various manye next-to-last particular final previous present nuf |
| AP\$ | determiner/pronoun, post-determiner, genitive | other's |
| AP+AP | determiner/pronoun, post-determiner, hyphenated pair | many-much |
| AT | article | the an no a every th' ever' ye |
| BE | verb "to be", infinitive or imperative | be |
| BED | verb "to be", past tense, 2nd person singular or all persons plural | were |
| BED* | verb "to be", past tense, 2nd person singular or all persons plural, negated | weren't |
| BEDZ | verb "to be", past tense, 1st and 3rd person singular | was |
| BEDZ* | verb "to be", past tense, 1st and 3rd person singular, negated | wasn't |
| BEG | verb "to be", present participle or gerund | being |

| | | |
|----------------|--|--|
| BEM | verb "to be", present tense, 1st person singular | am |
| BEM* | verb "to be", present tense, 1st person singular, negated | ain't |
| BEN | verb "to be", past participle | been |
| BER | verb "to be", present tense, 2nd person singular or all persons plural | are art |
| BER* | verb "to be", present tense, 2nd person singular or all persons plural, negated | aren't ain't |
| BEZ | verb "to be", present tense, 3rd person singular | is |
| BEZ* | verb "to be", present tense, 3rd person singular, negated | isn't ain't |
| CC | conjunction, coordinating | and or but plus & either neither nor yet 'n' and/or minus an' |
| CD | numeral, cardinal | two one 1 four 2 1913 71 74 637 1937 8 five three million 87-31 29-5 seven 1,119 fifty-three 7.5 billion hundred 125,000 1,700 60 100 six ... |
| CD\$ | numeral, cardinal, genitive | 1960's 1961's .404's |
| CS | conjunction, subordinating | that as after whether before while like because if since for than altho until so unless though providing once lest s'posin' till whereas whereupon supposing tho' albeit then so's 'fore |
| DO | verb "to do", uninflected present tense, infinitive or imperative | do dost |
| DO* | verb "to do", uninflected present tense or imperative, negated | don't |
| DO+PPSS | verb "to do", past or present tense + pronoun, personal, nominative, not 3rd person singular | d'you |
| DOD | verb "to do", past tense | did done |
| DOD* | verb "to do", past tense, negated | didn't |
| DOZ | verb "to do", present | does |

| | | |
|-----------------|--|--|
| | tense, 3rd person singular | |
| DOZ* | verb "to do", present tense, 3rd person singular, negated | doesn't don't |
| DT | determiner/pronoun, singular | this each another that 'nother |
| DT\$ | determiner/pronoun, singular, genitive | another's |
| DT+BEZ | determiner/pronoun + verb "to be", present tense, 3rd person singular | that's |
| DT+MD | determiner/pronoun + modal auxiliary | that'll this'll |
| DTI | determiner/pronoun, singular or plural | any some |
| DTS | determiner/pronoun, plural | these those them |
| DTS+BEZ | pronoun, plural + verb "to be", present tense, 3rd person singular | them's |
| DTX | determiner, pronoun or double conjunction | neither either one |
| EX | existential there | there |
| EX+BEZ | existential there + verb "to be", present tense, 3rd person singular | there's |
| EX+HVD | existential there + verb "to have", past tense | there'd |
| EX+HVZ | existential there + verb "to have", present tense, 3rd person singular | there's |
| EX+MD | existential there + modal auxiliary | there'll there'd |
| FW-* | foreign word: negator | pas non ne |
| FW-AT | foreign word: article | la le el un die der ein keine eine das las les Il |
| FW-AT+NN | foreign word: article + noun, singular, common | l'orchestre l'identite l'arcade l'ange l'assistance l'activite L'Universite l'independance L'Union L'Unita l'osservatore |
| FW-AT+NP | foreign word: article + noun, singular, proper | L'Astree L'Imperiale |
| FW-BE | foreign word: verb "to be", infinitive or imperative | sit |

| | | |
|------------------|--|---|
| FW-BER | foreign word: verb "to be", present tense, 2nd person singular or all persons plural | sind sunt etes |
| FW-BEZ | foreign word: verb "to be", present tense, 3rd person singular | ist est |
| FW-CC | foreign word: conjunction, coordinating | et ma mais und aber och nec y |
| FW-CD | foreign word: numeral, cardinal | une cinq deux sieben unam zwei |
| FW-CS | foreign word: conjunction, subordinating | bevor quam ma |
| FW-DT | foreign word: determiner/pronoun, singular | hoc |
| FW-DT+BEZ | foreign word: determiner + verb "to be", present tense, 3rd person singular | c'est |
| FW-DTS | foreign word: determiner/pronoun, plural | haec |
| FW-HV | foreign word: verb "to have", present tense, not 3rd person singular | habe |
| FW-IN | foreign word: preposition | ad de en a par con dans ex von auf super post sine sur sub avec per inter sans pour pendant in di |
| FW-IN+AT | foreign word: preposition + article | della des du aux zur d'un del dell' |
| FW-IN+NN | foreign word: preposition + noun, singular, common | d'etat d'hotel d'argent d'identite d'art |
| FW-IN+NP | foreign word: preposition + noun, singular, proper | d'Yquem d'Eiffel |
| FW-JJ | foreign word: adjective | avant Espagnol sinfonica Siciliana Philharmonique grand publique haute noire bouffe Douce meme humaine bel serieuses royaux anticus presto Sovietskaya Bayerische comique schwarzen ... |
| FW-JJR | foreign word: adjective, comparative | fortiori |
| FW-JJT | foreign word: adjective, superlative | optimo |

| | | |
|-------------------|---|---|
| FW-NN | foreign word: noun, singular, common | ballet esprit ersatz mano chatte goutte sang Fledermaus oud def kolkhoz roi troika canto boite blutwurst carne muzyka bonheur monde piece force ... |
| FW-NN\$ | foreign word: noun, singular, common, genitive | corporis intellectus arte's dei aeternitatis senioritatis curiae patronne's chambre's |
| FW-NNS | foreign word: noun, plural, common | al culpas vopos boites hafli's kolkhozes augen tyrannis alpha-beta-gammas metis banditos rata phis negociants crus Einsatzkommandos kamikaze wohaws sabinas zorrillas palazzi engages coureurs corroborées yori Übermenschen ... |
| FW-NP | foreign word: noun, singular, proper | Karshilama Dieu Rundfunk Afrique Espanol Afrika Spagna Gott Carthago deus |
| FW-NPS | foreign word: noun, plural, proper | Svenskarna Atlantes Dieux |
| FW-NR | foreign word: noun, singular, adverbial | heute morgen aujourd'hui hoy |
| FW-OD | foreign word: numeral, ordinal | 18e 17e quintus |
| FW-PN | foreign word: pronoun, nominal | hoc |
| FW-PP\$ | foreign word: determiner, possessive | mea mon deras vos |
| FW-PPL | foreign word: pronoun, singular, reflexive | se |
| FW-PPL+VBZ | foreign word: pronoun, singular, reflexive + verb, present tense, 3rd person singular | s'excuse s'accuse |
| FW-PPO | pronoun, personal, accusative | lui me moi mi |
| FW-PPO+IN | foreign word: pronoun, personal, accusative + preposition | mecum tecum |
| FW-PPS | foreign word: pronoun, personal, nominative, 3rd person singular | il |
| FW-PPSS | foreign word: pronoun, personal, nominative, not 3rd person singular | ich vous sie je |
| FW-PPSS+HV | foreign word: pronoun, personal, nominative, not | j'ai |

| | | |
|-----------------|--|---|
| | 3rd person singular + verb "to have", present tense, not 3rd person singular | |
| FW-QL | foreign word: qualifier | minus |
| FW-RB | foreign word: adverb | bas assai deja um wiederum cito velociter vielleicht simpliciter non zu domi nuper sic forsan olim oui semper tout despues hors |
| FW-RB+CC | foreign word: adverb + conjunction, coordinating | forisque |
| FW-TO+VB | foreign word: infinitival to + verb, infinitive | d'entretenir |
| FW-UH | foreign word: interjection | sayonara bien adieu arigato bonjour adios bueno tchalo ciao o |
| FW-VB | foreign word: verb, present tense, not 3rd person singular, imperative or infinitive | nolo contendere vive fermate faciunt esse vade noli tangere dites duces meminisse iuvabit gosaimasu voulez habla ksu'u'peli'afo lacheln miuchi say allons strafe portant |
| FW-VBD | foreign word: verb, past tense | stabat peccavi audivi |
| FW-VBG | foreign word: verb, present participle or gerund | nolens volens appellans seq. obliterans servanda dicendi delenda |
| FW-VBN | foreign word: verb, past participle | vue verstrichen rasa verboten engages |
| FW-VBZ | foreign word: verb, present tense, 3rd person singular | gouverne sinkt sigue diapiace |
| FW-WDT | foreign word: WH-determiner | quo qua quod que quok |
| FW-WPO | foreign word: WH-pronoun, accusative | quibusdam |
| FW-WPS | foreign word: WH-pronoun, nominative | qui |
| HV | verb "to have", uninflected present tense, infinitive or imperative | have hast |
| HV* | verb "to have", uninflected present tense or imperative, negated | haven't ain't |
| HV+TO | verb "to have", uninflected present tense + infinitival to | hafta |
| HVD | verb "to have", past tense | had |

| | | |
|---------------|---|---|
| HVD* | verb "to have", past tense, negated | hadn't |
| HVG | verb "to have", present participle or gerund | having |
| HVN | verb "to have", past participle | had |
| HVZ | verb "to have", present tense, 3rd person singular | has hath |
| HVZ* | verb "to have", present tense, 3rd person singular, negated | hasn't ain't |
| IN | preposition | of in for by considering to on among at through with under into regarding than since despite according per before toward against as after during including between without except upon out over ... |
| IN+IN | preposition, hyphenated pair | f'ovuh |
| IN+PPO | preposition + pronoun, personal, accusative | t'hi-im |
| JJ | adjective | recent over-all possible hard-fought favorable hard meager fit such widespread outmoded inadequate ambiguous grand clerical effective orderly federal foster general proportionate ... |
| JJ\$ | adjective, genitive | Great's |
| JJ+JJ | adjective, hyphenated pair | big-large long-far |
| JJR | adjective, comparative | greater older further earlier later freer franker wider better deeper firmer tougher faster higher bigger worse younger lighter nicer slower happier frothier Greater newer Elder ... |
| JJR+CS | adjective + conjunction, coordinating | lighter'n |
| JJS | adjective, semantically superlative | top chief principal northernmost master key head main tops utmost innermost foremost uppermost paramount topmost |
| JJT | adjective, superlative | best largest coolest calmest latest greatest earliest simplest strongest newest fiercest unhappiest worst youngest worthiest fastest hottest fittest lowest finest smallest staunchest ... |
| MD | modal auxillary | should may might will would must can could shall ought need wilt |
| MD* | modal auxillary, negated | cannot couldn't wouldn't can't won't shouldn't shan't mustn't musn't |
| MD+HV | modal auxillary + verb | shouldda musta coulda must've woulda could've |

| | | |
|----------------|---|---|
| | "to have", uninflected form | |
| MD+PPSS | modal auxillary + pronoun, personal, nominative, not 3rd person singular | willya |
| MD+TO | modal auxillary + infinitival to | oughta |
| NN | noun, singular, common | failure burden court fire appointment awarding compensation Mayor interim committee fact effect airport management surveillance jail doctor intern extern night weekend duty legislation Tax Office ... |
| NN\$ | noun, singular, common, genitive | season's world's player's night's chapter's golf's football's baseball's club's U.'s coach's bride's bridegroom's board's county's firm's company's superintendent's mob's Navy's ... |
| NN+BEZ | noun, singular, common + verb "to be", present tense, 3rd person singular | water's camera's sky's kid's Pa's heat's throat's father's money's undersecretary's granite's level's wife's fat's Knife's fire's name's hell's leg's sun's roulette's cane's guy's kind's baseball's ... |
| NN+HVD | noun, singular, common + verb "to have", past tense | Pa'd |
| NN+HVZ | noun, singular, common + verb "to have", present tense, 3rd person singular | guy's Knife's boat's summer's rain's company's |
| NN+IN | noun, singular, common + preposition | buncha |
| NN+MD | noun, singular, common + modal auxillary | cowhand'd sun'll |
| NN+NN | noun, singular, common, hyphenated pair | stomach-belly |
| NNS | noun, plural, common | irregularities presentments thanks reports voters laws legislators years areas adjustments chambers \$100 bonds courts sales details raises sessions members congressmen votes polls calls ... |
| NNS\$ | noun, plural, common, genitive | taxpayers' children's members' States' women's cutters' motorists' steelmakers' hours' Nations' lawyers' prisoners' architects' tourists' Employers' secretaries' Rogues' ... |
| NNS+MD | noun, plural, common + modal auxillary | duds'd oystchers'll |
| NP | noun, singular, proper | Fulton Atlanta September-October Durwood |

| | | |
|---------------|---|--|
| | | Pye Ivan Allen Jr. Jan. Alpharetta Grady William B. Hartsfield Pearl Williams Aug. Berry J. M. Cheshire Griffin Opelika Ala. E. Pelham Snodgrass ... |
| NP\$ | noun, singular, proper, genitive | Green's Landis' Smith's Carreon's Allison's Boston's Spahn's Willie's Mickey's Milwaukee's Mays' Howsam's Mantle's Shaw's Wagner's Rickey's Shea's Palmer's Arnold's Broglio's ... |
| NP+BEZ | noun, singular, proper + verb "to be", present tense, 3rd person singular | W.'s Ike's Mack's Jack's Kate's Katharine's Black's Arthur's Seaton's Buckhorn's Breed's Penny's Rob's Kitty's Blackwell's Myra's Wally's Lucille's Springfield's Arlene's |
| NP+HVZ | noun, singular, proper + verb "to have", present tense, 3rd person singular | Bill's Guardino's Celie's Skolman's Crosson's Tim's Wally's |
| NP+MD | noun, singular, proper + modal auxiliary | Gyp'll John'll |
| NPS | noun, plural, proper | Chases Aderholds Chapelles Armisteads Lockies Carbones French Marskmen Toppers Franciscans Romans Cadillacs Masons Blacks Catholics British Dixiecrats Mississippians Congresses ... |
| NPS\$ | noun, plural, proper, genitive | Republicans' Orioles' Birds' Yanks' Redbirds' Bucs' Yankees' Stevenses' Geraghtys' Burkes' Wackers' Achaeans' Dresbachs' Russians' Democrats' Gershwins' Adventists' Negroes' Catholics' ... |
| NR | noun, singular, adverbial | Friday home Wednesday Tuesday Monday Sunday Thursday yesterday tomorrow tonight West East Saturday west left east downtown north northeast southeast northwest North South right ... |
| NR\$ | noun, singular, adverbial, genitive | Saturday's Monday's yesterday's tonight's tomorrow's Sunday's Wednesday's Friday's today's Tuesday's West's Today's South's |
| NR+MD | noun, singular, adverbial + modal auxiliary | today'll |
| NRS | noun, plural, adverbial | Sundays Mondays Saturdays Wednesdays Souths Fridays |
| OD | numeral, ordinal | first 13th third nineteenth 2d 61st second sixth eighth ninth twenty-first eleventh 50th eighteenth- Thirty-ninth 72nd 1/20th twentieth mid-19th thousandth 350th sixteenth 701st ... |
| PN | pronoun, nominal | none something everything one anyone nothing nobody everybody everyone anybody anything |

| | | |
|----------------|---|--|
| | | someone no-one nothin |
| PN\$ | pronoun, nominal, genitive | one's someone's anybody's nobody's everybody's anyone's everyone's |
| PN+BEZ | pronoun, nominal + verb "to be", present tense, 3rd person singular | nothing's everything's somebody's nobody's someone's |
| PN+HVD | pronoun, nominal + verb "to have", past tense | nobody'd |
| PN+HVZ | pronoun, nominal + verb "to have", present tense, 3rd person singular | nobody's somebody's one's |
| PN+MD | pronoun, nominal + modal auxiliary | someone'll somebody'll anybody'd |
| PP\$ | determiner, possessive | our its his their my your her out thy mine thine |
| PP\$\$ | pronoun, possessive | ours mine his hers theirs yours |
| PPL | pronoun, singular, reflexive | itself himself myself yourself herself oneself |
| PPLS | pronoun, plural, reflexive | themselves ourselves yourselves |
| PPO | pronoun, personal, accusative | them it him me us you 'em her thee we'uns |
| PPS | pronoun, personal, nominative, 3rd person singular | it he she thee |
| PPS+BEZ | pronoun, personal, nominative, 3rd person singular + verb "to be", present tense, 3rd person singular | it's he's she's |
| PPS+HVD | pronoun, personal, nominative, 3rd person singular + verb "to have", past tense | she'd he'd it'd |
| PPS+HVZ | pronoun, personal, nominative, 3rd person singular + verb "to have", present tense, 3rd person singular | it's he's she's |
| PPS+MD | pronoun, personal, nominative, 3rd person singular + modal auxiliary | he'll she'll it'll he'd it'd she'd |
| PPSS | pronoun, personal, nominative, not 3rd person singular | they we I you ye thou you'uns |

| | | |
|------------------|--|--|
| PPSS+BEM | pronoun, personal, nominative, not 3rd person singular + verb "to be", present tense, 1st person singular | I'm Ahm |
| PPSS+BER | pronoun, personal, nominative, not 3rd person singular + verb "to be", present tense, 2nd person singular or all persons plural | we're you're they're |
| PPSS+BEZ | pronoun, personal, nominative, not 3rd person singular + verb "to be", present tense, 3rd person singular | you's |
| PPSS+BEZ* | pronoun, personal, nominative, not 3rd person singular + verb "to be", present tense, 3rd person singular, negated | 'tain't |
| PPSS+HV | pronoun, personal, nominative, not 3rd person singular + verb "to have", uninflected present tense | I've we've they've you've |
| PPSS+HVD | pronoun, personal, nominative, not 3rd person singular + verb "to have", past tense | I'd you'd we'd they'd |
| PPSS+MD | pronoun, personal, nominative, not 3rd person singular + modal auxillary | you'll we'll I'll we'd I'd they'll they'd you'd |
| PPSS+VB | pronoun, personal, nominative, not 3rd person singular + verb "to verb", uninflected present tense | y'know |
| QL | qualifier, pre | well less very most so real as highly fundamentally even how much remarkably somewhat more completely too thus ill deeply little overly halfway almost impossibly far severly such ... |
| QLP | qualifier, post | indeed enough still 'nuff |
| RB | adverb | only often generally also nevertheless upon |

| | | |
|---------------|---|--|
| | | together back newly no likely meanwhile near then heavily there apparently yet outright fully aside consistently specifically formally ever just ... |
| RB\$ | adverb, genitive | else's |
| RB+BEZ | adverb + verb "to be", present tense, 3rd person singular | here's there's |
| RB+CS | adverb + conjunction, coordinating | well's soon's |
| RBR | adverb, comparative | further earlier better later higher tougher more harder longer sooner less faster easier louder farther oftener nearer cheaper slower tighter lower worse heavier quicker ... |
| RBR+CS | adverb, comparative + conjunction, coordinating | more'n |
| RBT | adverb, superlative | most best highest uppermost nearest brightest hardest fastest deepest farthest loudest ... |
| RN | adverb, nominal | here afar then |
| RP | adverb, particle | up out off down over on in about through across after |
| RP+IN | adverb, particle + preposition | out'n outta |
| TO | infinitival to | to t' |
| TO+VB | infinitival to + verb, infinitive | t'jawn t'lah |
| UH | interjection | Hurrah bang whee hmpf ah goodbye oops oh- the-pain-of-it ha crunch say oh why see well hello lo alas tarantara rum-tum-tum gosh hell keerist Jesus Keeeerist boy c'mon 'mon goddamn bah hoo-pig damn ... |
| VB | verb, base: uninflected present, imperative or infinitive | investigate find act follow inure achieve reduce take remedy re-set distribute realize disable feel receive continue place protect eliminate elaborate work permit run enter force ... |
| VB+AT | verb, base: uninflected present or infinitive + article | wanna |
| VB+IN | verb, base: uninflected present, imperative or infinitive + preposition | lookit |
| VB+JJ | verb, base: uninflected present, imperative or infinitive + adjective | die-dead |

| | | |
|-------------------|---|--|
| VB+PPO | verb, uninflected present tense + pronoun, personal, accusative | let's lemme gimme |
| VB+RP | verb, imperative + adverbial particle | g'ahn c'mon |
| VB+TO | verb, base: uninflected present, imperative or infinitive + infinitival to | wanta wanna |
| VB+VB | verb, base: uninflected present, imperative or infinitive; hyphenated pair | say-speak |
| VBD | verb, past tense | said produced took recommended commented urged found added praised charged listed became announced brought attended wanted voted defeated received got stood shot scheduled feared promised made ... |
| VBG | verb, present participle or gerund | modernizing improving purchasing Purchasing lacking enabling pricing keeping getting picking entering voting warning making strengthening setting neighboring attending participating moving ... |
| VBG+TO | verb, present participle + infinitival to | gonna |
| VCN | verb, past participle | conducted charged won received studied revised operated accepted combined experienced recommended effected granted seen protected adopted retarded notarized selected composed gotten printed ... |
| VCN+TO | verb, past participle + infinitival to | gotta |
| VBZ | verb, present tense, 3rd person singular | deserves believes receives takes goes expires says opposes starts permits expects thinks faces votes teaches holds calls fears spends collects backs eliminates sets flies gives seeks reads ... |
| WDT | WH-determiner | which what whatever whichever whichever-the-hell |
| WDT+BER | WH-determiner + verb "to be", present tense, 2nd person singular or all persons plural | what're |
| WDT+BER+PP | WH-determiner + verb "to be", present, 2nd person singular or all persons plural + pronoun, personal, nominative, not | whaddya |

| | | |
|-------------------|--|---|
| | 3rd person singular | |
| WDT+BEZ | WH-determiner + verb "to be", present tense, 3rd person singular | what's |
| WDT+DO+PPS | WH-determiner + verb "to do", uninflected present tense + pronoun, personal, nominative, not 3rd person singular | whaddya |
| WDT+DOD | WH-determiner + verb "to do", past tense | what'd |
| WDT+HVZ | WH-determiner + verb "to have", present tense, 3rd person singular | what's |
| WP\$ | WH-pronoun, genitive | whose whosever |
| WPO | WH-pronoun, accusative | whom that who |
| WPS | WH-pronoun, nominative | that who whoever whosoever what whatsoever |
| WPS+BEZ | WH-pronoun, nominative + verb "to be", present, 3rd person singular | that's who's |
| WPS+HVD | WH-pronoun, nominative + verb "to have", past tense | who'd |
| WPS+HVZ | WH-pronoun, nominative + verb "to have", present tense, 3rd person singular | who's that's |
| WPS+MD | WH-pronoun, nominative + modal auxiliary | who'll that'd who'd that'll |
| WQL | WH-qualifier | however how |
| WRB | WH-adverb | however when where why whereby wherever how whenever whereon wherein wherewith wheare wherefore whereof howsabout |
| WRB+BER | WH-adverb + verb "to be", present, 2nd person singular or all persons plural | where're |
| WRB+BEZ | WH-adverb + verb "to be", present, 3rd person singular | how's where's |
| WRB+DO | WH-adverb + verb "to do", present, not 3rd person singular | howda |
| WRB+DOD | WH-adverb + verb "to do", past tense | where'd how'd |

| | | |
|-----------------|--|---------|
| WRB+DOD* | WH-adverb + verb "to do", past tense, negated | whyn't |
| WRB+DOZ | WH-adverb + verb "to do", present tense, 3rd person singular | how's |
| WRB+IN | WH-adverb + preposition | why'n |
| WRB+MD | WH-adverb + modal auxillary | where'd |

Apêndice B

Lista de etiquetas do *Brown Corpus* incluindo as etiquetas compostas

Neste anexo apresenta-se a lista de todas as etiquetas incluindo as etiquetas compostas.

at nptl nntl jjtl vbd nr nn in np\$ jj “ ” cs dti nns . rbr , wdt hvd vbz cc intl bedz vbn np ben to
vb rb dt pps dod ap ber hv dts vbg ppo ql jjt abx nnhl vbnhl wrb cd md be jjr vbgtl bez nn\$tl hvz
abn pn ppss pp\$ do nn\$ nnshl wps * ex vbhl : () nntsl nps jjs rp bed od beg athl vbghl attl ppl
doz nphl nr\$ dod* bedz* ,hl cctl md* nns\$ ppss+ber ’ ppss+bem cdtl rbt (hl)hl mdhl vbzhl inhl
jjhl ppls cdhl wpo jjstl abl berhl pps+hvz vbdhl rphl md*hl aphl cshl dt\$ hvn fwin fwdt vbntl nrtl
nns\$tl fwnn hvg dtx odtl bem rbhl ppss+md npshl nps\$ wp\$ nntlhl cchl pps+bez aptl uhtl bezhl
tohl do* vbntlhl nntslhl dthl behl doz* qlp jjrhl ppss+hvd fwin+nn pp\$\$ jjthl nptlhl npstl md+hv
np\$tl odhl jjrtl vbdtl dt+bez ex+bez ppss+hv :hl pps+md uh fwcc fwnnns bedzhl nn\$hl hvd* bez*
ap\$ np+bez fwattl vbtl rbtl mdtl pn+hvz fwjjtl fwnntl abnhl .hl pps+bezhl nrhl hvdhl rb\$ fwathl
dohl pp\$tl fwintl wps+bez *hl dtihl pnhl cd\$ ber* nns\$hl pn\$ bertl totl fwjj bed* rb+bez vb+ppo
ppsshl hvz* fwin+nntl fwin+attl nnnc jjnc nr\$tl fwpp\$nc fwvb fwvbnc jjrnc nps\$tl qltl fwat fw* fwcd
wql fwwdt wdt+bez ppohl jjtlhl nps\$hl uhhl wrbhl wdt+bezhl np\$hl wdthl dozhhl rbnc hl pp\$hl :tl
rn vbztl fwppl wptl nrs beztl ppotl abntl fwnp pplhl ppsstl innnc fwvbz fwppo dotl qlhl dttl fwnnstl
fwvbdtl wrb+beztl wrbtl wqltl fwcctl fwat+nptl fwodtl fwat+nntl nil fwin+att ppss+bertl wpstl
hvztl bentl rptl cdtlhl betl intlhl hv* cstl ppss+hvtdl dod*tl ppstl fwrbtl pntl wdt+beztl nrtlhl cctlhl
attlhl fwjjnc fwjjr fwnn\$ fwvbd ,tl hvhl vb+in wdt+ber+pp vbg+to nn+hvz vbn+to fwnn\$tl fw*tl
nn+bez fwuhnc do*hl ppshl fwpp\$ rb+bezhl jjshl dtshl nnsnc ppssnc ppsnc pnnc fwrb fwvbn ppltl
wdt+hvz wrb+bez wps+md npnc fwin+nptl vbnnnc atnc fwnnnnc nn+md jjr+cs fwnnnsnc pps+hvd
vb+rp nrstl fwbez fwber fwvbg fwnrtl fwppss fwhv fwnptl vbnc doz*tl fwnpstl fwpp\$tl fwcs *tl hvtl

fwpn fwppo+in dtnc jjttl dtitl .tl ex+md fwuh fwnps fwcdtl fwbe fwpps fwnr fwto+vb fwin+at jj\$tl
fwvbt1 fwrb+cc fwwpo fwnntlnc fwwps fwppl+vbz doztl vbntlnc nntlnc nntlnc fwdts nns\$tlhl fw-
vbgtl exhl pponc ppss+bern wrbnc bernc tonc nrnc uhnc hvnc ,nc wdt+beznc .nc np+hvznc hvznc
cdnc qlnc wps+beznc pp\$nc dt+beznc rb+beznc wdtnc bedznc vbdnc rpnc vbznc vbgnc ppss+mdnc
*nc exnc ber*nc apnc donc bednc ccnc pps+beznc mdnc ppss+bernc csnc nns\$nc pplnc rbrnc beznc
odnc np+beznc abnnc wpsnc jjtnc dodnc wponc bemnc npsnc nn+nnnc jj+jjnc ap+apnc vb+jjnc
vb+vbnc wps1 fwql jjtlnc fwjjt wps+bezt1 hvghl fwdt+bez ex+hvd wps+hvz pn+md vb+to dt+md
hv+to md+to md+ppss nr+md nn+in rp+in bem* pn+bez wps+hvd nn+hvdt1 wdt+dod nn+bezt1
wrb+do wrb+in np+hvz wrb+dod wrb+md ex+hvz nn+hvzt1 ppss+vb wrb+ber nns+md ppss+bez
ppss+bez* rbr+cs np+md to+vb in+ppo in+in do+ppss vb+at wrb+doz dts+bez wdt+do+pps
rb+cs wrb+dod* wdt+ber fwodnc fwppss+hv pn+hvd fwuht1

Apêndice C

Ficheiro wlistnorarenonum- brown_train_dev_test.txt

Neste anexo apresenta-se parte do ficheiro wlistnorarenonumbrown_train_dev_test.txt. Na primeira coluna, para uma melhor compreensão, embora não fazendo parte deste ficheiro, apresenta-se a ordem da palavra no ficheiro. O ficheiro corresponde à segundas e terceira colunas aqui apresentadas. Na segunda coluna aparece a palavra e na terceira coluna o número de ocorrências da palavra nos textos que constituem o *Brown Corpus*.

1 ! 1596 2 \$*number* 545

3 \$.*number* 21

4 & 163

5 ' 317

6 " 8789

7 '*number* 30

8 'em 41

9 'im 6

10 'round 5

11 (2435

12) 2466

13 **ya 5

14 **yc 8

15 **yf 8

16 **yl 6

17 **yt 4

18 **zg 32

19 *UNKNOWN* 54129

20 *number* 6939

21 , 58334

22 - 78

23 – 3432

24 . 49347

25 / 20

26 : 1795

27 :*number* 1

28 ; 5566

29 ? 4693

30 A 1314

31 A's 6

32 A. 110

33 A.D. 10

34 A.L.A.M. 13

35 A.M. 13

36 A.M.A. 4

37 ABO 5

38 ADC 8

39 AFL-CIO 5

40 AIA 10

41 AID 4

42 AIMO 5

43 AM 9

44 AP 14

45 AWOC 5

46 Aaron 7

47 Abbey 6

48 Abel 20

49 About 49

50 Above 21

51 Abraham 6
52 Abstract 4
53 Academy 18
54 Acala 5
55 Accacia 4
56 According 29
57 Accordingly 19
58 Achievement 8
59 Acropolis 6
60 Across 8
61 Act 116
62 Acting 9
63 Action 4
64 Active 6
65 Activities 5
66 Actually 38
67 Ada 17
68 Ada's 4
69 Adam 44
70 Adams 42
71 Add 9
72 Additional 14
73 Adelia 5
74 Adenauer 4
75 Adios 6
76 Adjusted 10
77 Adlai 7
78 Administration 58
79 Administrative 4
80 Ado 4
81 Adolf 4
82 Adrian 4
83 Advisory 14
84 Advocate 5

85 Aegean 15
86 Aerospace 4
87 Aeschylus 4
88 Af 995
89 Af-stage 4
90 Affairs 22
91 Africa 45
92 African 28
93 Africans 4
94 Afro-Asian 4
95 After 246
96 Afterwards 4
97 Again 43
98 Against 9
99 Age 12
100 Agency 8
101 Ages 14
102 Agnese 6
103 Agreement 26
104 Agriculture 11
105 Ah 18
106 Aid 6
107 Aids 5
108 Ain't 4
109 Air 39
110 Aircraft 6
111 Airport 8
112 Aj 118
113 Al 15
114 Ala. 7
115 Alabama 20

.
.
.

17542 yet 283
17543 yield 35
17544 yielded 12
17545 yielding 9
17546 yields 7
17547 you 2766
17548 you'd 28
17549 you'll 60
17550 you're 93
17551 you've 43
17552 young 359
17553 younger 40
17554 youngest 12
17555 youngster 8
17556 youngsters 16
17557 your 868
17558 yours 24
17559 yourself 67
17560 yourselves 8
17561 youth 73
17562 youthful 12
17563 youths 10
17564 yow 5
17565 yrs. 4
17566 zeal 8
17567 zealous 4
17568 zero 23
17569 zest 5
17570 zigzagging 4
17571 zinc 9
17572 zing 6
17573 zone 10
17574 zoning 5
17575 zoo 6
